

```

%This Worksheet simulates finding the nonlinear regression exponential model
%without data linearization
function varargout = mtl_reg_sim_complinearized()
%Input arrays X and Y. Also give a reasonable initial value of b.
X=[0.5,1,3,5,7,9];
Y=[1.648,2.71,150.08,148.4,1096,8103];
initial_value_b=0.1;
% End of input
clc

disp(sprintf('Nonlinear Regression without Data Linearization'))
disp(sprintf('©2007 Autar Kaw'))
disp(sprintf('University of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu'))

disp(sprintf('\n\nNOTE: This worksheet demonstrates the use of Matlab to illustrate'))
disp(sprintf('the procedure to regress a given data set to a nonlinear exponential model without'))
disp(sprintf('data linearization, that is,  $y=a*\exp(b*x)$ '))
%-----
disp(sprintf('\n\n***** Introduction *****'))

disp(sprintf('Using the sum of the square of residuals, and minimizing with respect '))
disp(sprintf('to (a) and (b) in will give two simultaneous nonlinear equations'))
disp(sprintf('Fortunately, the two simultaneous linear equations can be reduced to'))
disp(sprintf('one nonlinear equation in (b). Once solved, a can be found directly'))
disp(sprintf('The input arrays X and Y:'))
X
Y
%----- Exponential-----
disp(sprintf('\n\n***** Exponential *****'))

disp(sprintf('Given (x1,y1),(x2,y2),...(xn,yn), best fit  $y=a*\exp(b*x)$  to the data.))

n=length(X);
Z=zeros(1,n);
%Finding the value of (b) first. See the function eqn
b_value=fzero(@(b) f(b,X,Y,n),initial_value_b);

%Call fzero with a one-argument anonymous function that captures that value of a and calls myfun with
two arguments:
%Calculating the value of (a)
sum_yex=0;
sum_e2x=0;
for i=1:n
    sum_yex=sum_yex+Y(i)*exp(b_value*X(i));
    sum_e2x=sum_e2x+exp(2*b_value*X(i));
end
a_value=sum_yex/sum_e2x;
% Writing out the values of (a) and (b)
a_value
b_value
disp(sprintf('\n\nThe model without data linearization is described as\n
y=%5g',a_value))
disp(sprintf('\b*exp(%5g',b_value))

```

```

disp(sprintf('\b*x)          (4)'))

% Now doing the model with data linearization
Z=zeros(1,n);
for i=1:n
    Z(i)=log(Y(i));
end
Z;
xav=sum(X)/n;
zav=sum(Z)/n;
sum(Z);
Sxz=0;
Sxx=0;
for i=1:n
    Sxz=Sxz + X(i)*Z(i)-xav*zav;
    Sxx=Sxx + (X(i))^2-xav^2;
end

Sxx;
Sxz;

a1=Sxz/Sxx
a0=zav-a1*xav
disp(sprintf('Now since a0 and a1 are found, the original constants with the model'))
disp(sprintf('are found as'))
a=exp(a0)
b=a1

disp(sprintf('\n\nThe model with data linearization is described as\n          y=%5g',a))
disp(sprintf('\b*exp(%5g',b))
disp(sprintf('\b*x)          (4)'))
disp(sprintf('\n\nThe model without data linearization is described as\n          y=%5g',a_value))
disp(sprintf('\b*exp(%5g',b_value))

disp(sprintf('\b*x)          (4)'))

%Plotting the graph of x*exp(b*x) and the data points.
xp=(0:0.001:max(X));
yplin=zeros(1,length(xp));
ypwithout=zeros(1,length(xp));
for i=1:length(xp)
ypwithout(i)=a_value*exp(b_value*xp(i));
yplin(i)=a*exp(b*xp(i));
end
plot(xp,ypwithout,'b')
xlabel('x')
ylabel('y=a*exp(b*x)')
hold on
plot(xp,yplin,'r')
hold on
plot(X,Y,'bo','MarkerFaceColor','b')
hold off
end
% This is the function where you set up the left hand side of the equation f(b)=0
% Using fzero the root of the equation is found

```

```
function eqn1 = f(b,X,Y,n)
sum_yxex=0;
sum_yex=0;
sum_ex2=0;
sum_xex2=0;
for i=1:n
    sum_yxex=sum_yxex+Y(i)*X(i)*exp(b*X(i));
    sum_yex=sum_yex+Y(i)*exp(b*X(i));
    sum_ex2=sum_ex2+exp(2*b*X(i));
    sum_xex2=sum_xex2+X(i)*exp(2*b*X(i));
end
eqn1 = sum_yxex-sum_yex/sum_ex2*sum_xex2;
end
```